REMARKS

Receipt of the Office Action of October 3, 2008 is gratefully acknowledged.

Claims 5 - 8 have been examined with the following result: claims 6 - 8 are rejected under 35 USC 112, second paragraph as indefinite; and claims 5 - 8 are rejected under 35 USC 102(a) by "PROFIBUS technology and application-system description" Oct. 2002," hereinafter PROFIBUS.

Rejection under 35 USC 112, second paragraph

Regarding claim 6, the examiner states that "[t]here is no description of FDT/DTM in the specification." Regarding claim 7, the examiner states that "[t]here is no description of PDM and HCF in the specification." Regarding claim 8, the examiner states that "[t]here is no description of EDD in the specification."

As the examiner is aware FDT/DTM, PDM, HCF and EDD are acronyms for well known technical concepts in wide use in automation, for example. In fact, there identity can be easily obtained from the internet. For example, submitted herewith is a Wikipedia discussion of PDM as well as a definition of FDT/DTM. Without more, one can understand what PDM and FDT/DTM mean, and, moreover, in the context of the present invention. In addition, the PROFIBUS catalogue cited by the examiner, includes a discussion of certain of these acronyms.

Perhaps the problem arises because of the inclusion in claims 6 and 8 of the "PROFIBUS Guideline -Order No. 2.162." This reference has been deleted from claims 6 and 8 and retained in the specification only. The recitation in the specification notes that the disclosure of this "Guideline" has been incorporated by reference. Certainly, those skilled in the art should have no difficulty in

understanding what the claims mean when viewed in conjunction with the acronyms. See, ***Morton International, Inc. v. Cardinal Chemical Co.***, 28 USPQ2d 1190 (Fed. Cir. 1993).

It is respectfully submitted that the use of the noted acronyms does not render claims 6 - 8 indefinite because they are well know.

Rejection under 35 USC 102(a)

The rejection under 35 USC 102(a) is respectfully traversed because the noted PROFIBUS reference does not contain every positively recited element of the claims. Relative to claim 5 the examiner refers us to page 27 of PROFIBUS and specifically to the passage headed by "Device Description as Software Component," which states that "[t]he DTM is generated by the device manufacturer and is included in delivery of the service." Where in this passage is there a disclosure of "converting the standard device descriptions by means of a compiler into corresponding software modules? It is not there. What is there is nothing more than what is stated in the background portion of the specification.

The examiner does refer to the passage under "DTM generation" which includes "generation from an existing device description using a compiler or interpreter." This passage is not sufficiently specific. Does it mean that standard device descriptions syntactically and semantically correct are produced? There is no reason to believe that this occurs.

The passage referred to by the examiner at page 27 of PROFIBUS with the heading 7.2 EDD, amounts to nothing more than a broad statement that EDD can be used rather than GSD. What is the "relevant **EDD file**?"

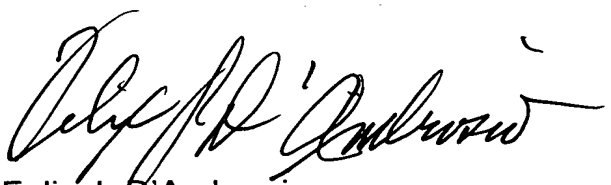
For 35 USC 102 to apply, the disclosure relied upon must be more specific.

The all elements rule requires a clear and convincing disclosure. Such a disclosure is believed to be lacking here.

In view of the foregoing, reconsideration and re-examination are respectfully requested and claims 5 - 8 found allowable.

Respectfully submitted,
BACON & THOMAS, PLLC

Date: April 3, 2008



Felix J. D'Ambrosio
Attorney for Applicant
Registration Number 25,721

Customer Number *23364*
BACON & THOMAS, PLLC
625 Slaters Lane, Fourth Floor
Alexandria, Virginia 22314
Telephone: (703) 683-0500
Facsimile: (703) 683-1080
S:\Producer\fjd\CLIENTS\Endress+Hauser Holding GmbH\WITT3005-CD0176\Prelim Amend.wpd

Product Data Management

From Wikipedia, the free encyclopedia

Product Data Management (PDM) is a category of computer software used to control data related to products. PDM creates and manages relations between sets of data that define a product, and store those relationships in a database. It is an important tool in product lifecycle management.

Contents

Introduction

Product Data Management (PDM) is focused on information relative to core operations of a product-based company. PDM is one element of a larger evolving field of enterprise master data - customer master, supplier master, employee master, etc.

Within PDM the focus is on managing and tracking the creation, change and archive of all information related to a product. The information being stored and managed (on one or more file servers) will include engineering data such as Computer-aided design (CAD) models, drawings and their associated documents.

"Associated documents" is a collector term for: requirements, specifications, manufacturing plans, assembly plans, test plans and test procedures. The package may also include product visualization data.

The central database will also manage metadata such as owner of a file and release status of the components. The package will: control check-in and check-out of the product data to multi-user; carry out engineering change management and release control on all versions/issues of components in a product; build and manipulate the product structure bill of materials (BOM) for assemblies; and assist in configurations management of product variants.

This enables automatic reports on product costs, etc. Furthermore, PDM enables companies producing complex products to spread product data into the entire PLM launch-process. This significantly enhances the effectiveness of the launch process.

Product Data Management is focused on capturing and maintaining information on products and/or services through its development and useful life. Typical information managed in the PDM module include

- Part number

communication. These rules apply to different layers of sophistication such as which physical connections to use, how hosts listen, how to interrupt, how to say good-bye,in short how to communicate, what language to use and many others. These rules, or protocols, that work together to ensure successful communication are groups into what is known as a protocol suite.

Object-oriented programming has extended the use of the term to include the programming protocols available for connections and communication between objects.

Generally, only the simplest protocols are used alone. Most protocols, especially in the context of communications or networking, are layered together into protocol stacks where the various tasks listed above are divided among different protocols in the stack.

Whereas the protocol stack denotes a specific combination of protocols that work together, a reference model is a software architecture that lists each layer and the services each should offer. The classic seven-layer reference model is the OSI model, which is used for conceptualizing protocol stacks and peer entities. This reference model also provides an opportunity to teach more general software engineering concepts like hiding, modularity, and delegation of tasks. This model has endured in spite of the demise of many of its protocols (and protocol stacks) originally sanctioned by the ISO. The *OSI model* is not the only reference model however.

Common protocols

- IP (Internet Protocol)
- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- HTTP (Hypertext Transfer Protocol)
- FTP (File Transfer Protocol)
- Telnet (Telnet Remote Protocol)
- SSH (Secure Shell Remote Protocol)
- POP3 (Post Office Protocol 3)
- SMTP (Simple Mail Transfer Protocol)
- IMAP (Internet Message Access Protocol)

See also

- Internet protocol suite
- Communications protocol
- List of network protocols
- Application programming interface
- Calling convention

1 February 2006

Defining FDT/DTM

Regarding the August 2005 *InTech* Networking & Communications by Craig McIntyre, "A single, open-standard field device," I have this to say: The title is a little bit misleading because FDT is not a standard for field devices. FDT/DTM is software. FDT/DTM is a field device tool / device type manager.

FDT/DTM itself does not communicate with intelligent devices. The communication takes place using existing bus technologies such as HART and Profibus. DTM interprets and displays the data. Electronic device description language (EDDL) and DTM do the same thing, but differently. DTM does not need EDDL and vice versa, therefore many consider them competing. Obviously one does not render the other or the instruments obsolete. However, they do compete, just like Foundation fieldbus and Profibus. You can't say Foundation fieldbus and Profibus complement each other.

EDDL also covers remote input/output and drives.

It is true both systems and instruments need to support both. Just as control systems today have interfaces for both fieldbus and Profibus, device management software will have to support both DTM and EDDL so any device can work regardless of the files it comes with. As well, a device needs to come with both EDDL and DTM so it can connect to any system.

Technically, a DTM can interpret device descriptions, but users are accepting it very well. They want native DTMs.

Jonas Berge, engineer
SMAR Singapore Pte. Ltd.